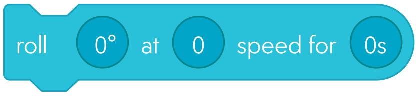
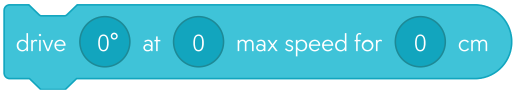
Block Library

Movements



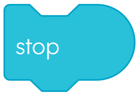
```
roll 0° at 0 speed for 0s
```

Roll: Combines heading, speed, and duration to make the robot roll.



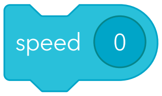
```
drive 0° at 0 max speed for 0 cm
```

Drive: combines heading, speed, and distance to make the robot drive.



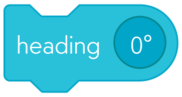
```
stop
```

Stop: Sets the speed to 0 to stop the robot when a Speed block is used.



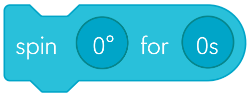
```
speed 0
```

Speed: Sets the target speed of the robot from -255 to 255. Positive speed is forward, negative speed is backward, and 0 is stopped.



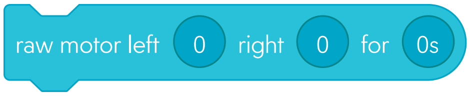
```
heading 0°
```

Heading: Sets the target direction the robot rolls. Assuming the robot is aimed with the blue tail light facing you, then 0° is forward, 90° is right, 270° is left, and 180° is back.



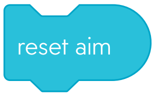
```
spin 0° for 0s
```

Spin: Spins the robot for a given number of degrees over time, where 360° is a single rotation.



```
raw motor left 0 right 0 for 0s
```

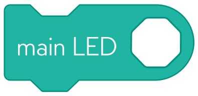
Raw Motor: This command disables the stabilization and will cause the robot to jump when both motors are set to full power. Controls the electrical power sent to the left and right motors independently, from -255 to 255, for a duration of seconds.



```
reset aim
```

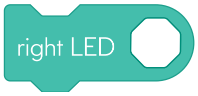
Reset Aim: Resets the heading calibration (aim) angle to use the current front-facing direction of the robot as 0°.

Lights



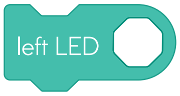
```
main LED
```

Main LED: Changes the color of the main LEDs. Set this using the color wheel and brightness slider, or the exact RGB (red, green, blue) values, from 0 to 255.



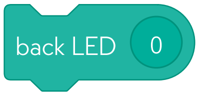
```
right LED
```

Right LED: Changes the color of the right LED. Set this using the color wheel and brightness slider or the exact RGB (red, green, blue) values, from 0 to 255.

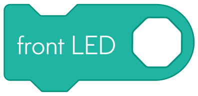


```
left LED
```

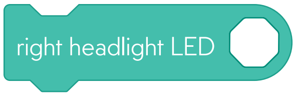
Left LED: Changes the color of the left LED. Set this using the color wheel and brightness slider or the exact RGB (red, green, blue) values, from 0 to 255.



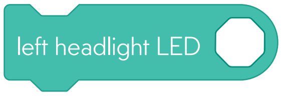
Back LED: Changes the color of the back LED. Turns on the back LED. This is limited to blue only. You can adjust the brightness by tapping on the number.



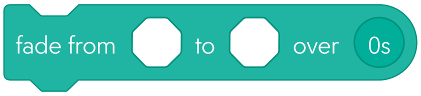
Front LED: Changes the color of the front LED. Set this using the color wheel and brightness slider, or the exact RGB (red, green, blue) values, from 0 to 255.



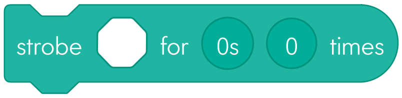
Right Headlight LED: Changes the color of the right headlight LED. Set this using the color wheel and brightness slider or the exact RGB (red, green, blue) values, from 0 to 255.



Left Headlight LED: Changes the color of the left headlight LED. Set this using the color wheel and brightness slider or the exact RGB (red, green, blue) values, from 0 to 255.

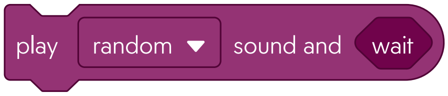


Fade: Changes the main LEDs from one color to another, for a duration of seconds. This command works as a standalone command and cannot be used to fade main LEDs while using a roll command.

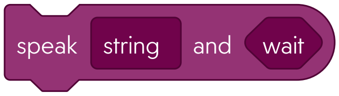


Strobe: Blinks the main LEDs for a period of seconds (that includes lights ON and OFF) and a count of cycles. A short period will produce a fast blink whereas a long period will produce a slow blink.

Sounds

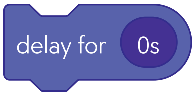


Sound: Plays a sound from your programming device (not from the robot) that you select from the list. Toggle the Randomize option to generate a random sound or select one from Sphero's library.

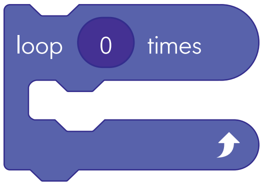


Speak: Speak strings (numbers and words) from your programming device. Type what you want your robot to say.

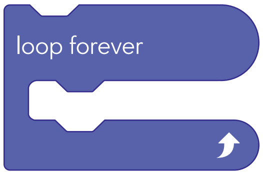
Controls



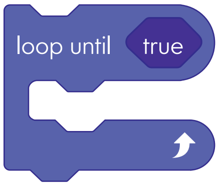
Delay: Delays execution of the next block for a number of seconds.



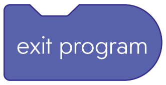
Loop: Repeats the blocks within for the number of loops specified, also known as a "for" loop.



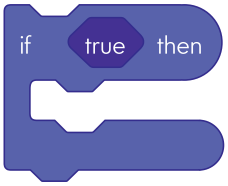
Loop Forever: Repeats the blocks within forever, also known as a "while 1" loop. You can use this to constantly evaluate conditions.



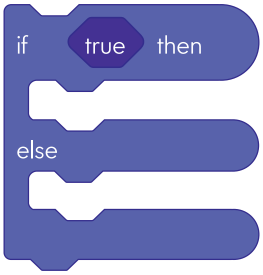
Loop Until: Repeats the blocks within until a condition is met, also known as a "while" loop. Drag any comparator into the "true" field to create a condition. Use "and" and "or" to combine or exclude conditions, which must be added before other comparators.



Exit Program: Stops all code and ends the program; the same as hitting the Stop button.

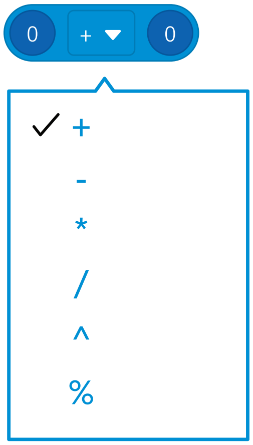


If Then: Calls the "if" blocks if the given condition is true. Use "and" and "or" to combine or exclude conditions, which must be added before other comparators.

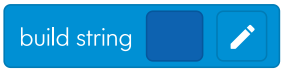


If Then Else: Calls the "if" blocks if the given condition is true; otherwise calls the "else" blocks. Use "and" and "or" to combine or exclude conditions, which must be added before other comparators.

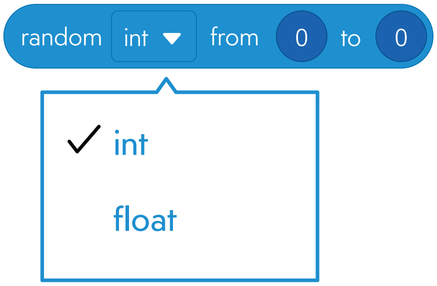
Operators



Basic Operators: Adds a mathematical action to calculate or evaluate a set of data or given value.

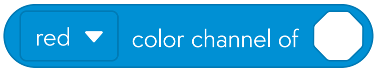


Build String: Combines multiple values into a single string. Those values can be numbers (variables, parameters, or sensors), strings, booleans, or colors.

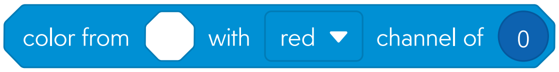


Random Int: Generates a random integer value (nearest whole number) within the given minimum and maximum.

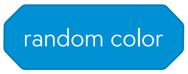
Random Float: Generates a random float value (including decimals) within the given minimum and maximum.



Color Channel: Gets the value of the red, green, or blue color channel for a given RGB value.

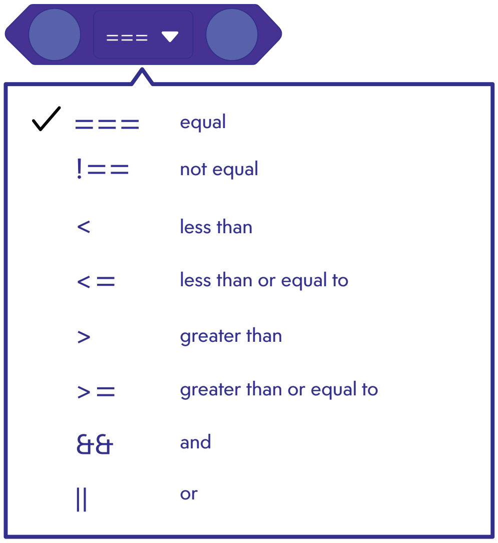


Color Mixer: Returns a new color by modifying a single channel (red, green, blue) of a given color.



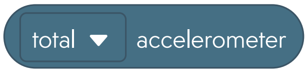
Random Color: Sets a random color when placed in a color block.

Comparators



Comparators: Comparators are blocks of code that compare two things (Sensor data, operators, strings, etc.). Comparators allow us to create logical checks, which, when "true," will allow the program to perform a certain action or event. If a comparison is true and logical, then the code will be executed.

Sensors



Accelerometer - Total: The accelerometer sensor measures the change in force in the robot.

- Total: The combined vector acceleration of all three axes, from of 0 to 14 g's.
- X-Axis: The left-to-right acceleration, from -8 to 8 g's.
- Y-Axis: The forward-to-back acceleration, from -8 to 8 g's.
- Z-Axis: The up-and-down acceleration, from -8 to 8 g's.

pitch ▾ orientation

Orientation: The orientation sensor measures the orientation of the robot.

- Pitch: The forward or backward tilt angle, from -180° to 180°.
- Roll: The left or right tilt angle, from -90° to 90°.
- Yaw: The spin (twist) angle, from -180° to 180°.

pitch ▾ gyroscope

Gyroscope: The gyroscope sensor measures the rate of spin in the robot in degrees per second.

- Pitch: The rate of forward or backward spin, from -2,000° to 2,000° per second.
- Roll: The rate of left or right spin, from -2,000° to 2,000° per second.
- Yaw: The rate of sideways spin, from -2,000° to 2,000° per second.

total ▾ velocity

Velocity - Total: The velocity sensor measures the speed in centimeters per second.

- Velocity Total: The combined vector speed of both axes which will always be a positive value, in centimeters per second.
- Velocity - X: The right (+) or left (-) speed, in centimeters per second.
- Velocity - Y: The forward (+) or back (-) speed, in centimeters per second.

total ▾ location

Location - Total: The distance from the location of the program start, which will always be a positive value, in centimeters.

distance

Distance: The total distance traveled, in centimeters.

speed

Speed: Target speed value, from -255 to 255.

heading

Heading: Target directional angle of the robot. Assuming you aim the robot with the blue tail light facing you, then 0° heading is forward, 90° is right, 180° is backward, and 270° is left.

main LED

Main LED: The RGB color of the main LEDs, from 0 to 255 for each color channel.

time elapsed

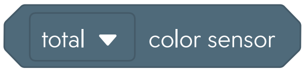
Time Elapsed: The amount of time with which the program has run (in seconds).

luminosity

Luminosity: The light intensity from 0 - 100,000 lux, where 0 lux is full darkness and 30,000-100,000 lux is direct sunlight.

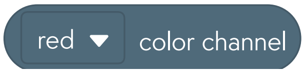
last message recieved

Luminosity: Returns which channel the last infrared message was received on.



total ▾ color sensor

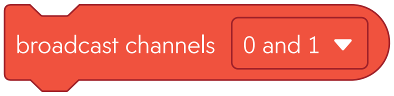
Color Sensor: Color value from 0 to 255 for all channels (red, green, blue) of the color sensor.



red ▾ color channel

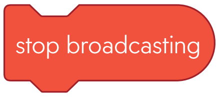
Color Channel: Number value from 0 to 255 for the selected color channel (red, green, or blue) of the color sensor.

Communications



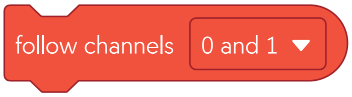
broadcast channels 0 and 1 ▾

Broadcast: Sets the IR emitters to broadcast on two specified channels, from 0 to 7. The broadcaster uses two channels so the following or evading BOLTs can detect these messages on their IR receivers with a sense of relative proximity to the broadcaster. You can't use a channel for more than one purpose at a time.



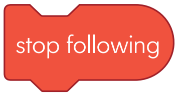
stop broadcasting

Stop Broadcast: Stops the broadcasting behavior.



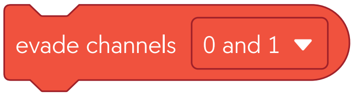
follow channels 0 and 1 ▾

Follow: Sets the IR receivers to look for broadcasting BOLTs on the same channel pair, from 0 to 7. Upon receiving messages from a broadcasting BOLT, the follower will adjust its heading and speed to follow the broadcaster.



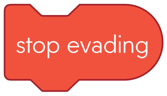
stop following

Stop Following: Stops the following behavior.



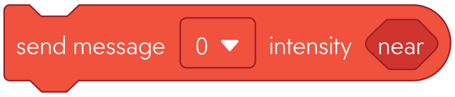
evade channels 0 and 1 ▾

Evade: Sets the IR receivers to look for broadcasting BOLTs on the same channel pair, from 0 to 7. Upon receiving messages from a broadcasting BOLT, the evader will adjust its heading to roll away from the broadcaster.



stop evading

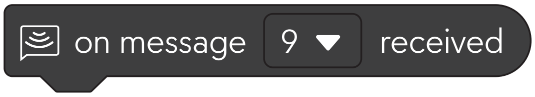
Stop Evading: Stops the evading behavior.



send message 0 ▾ intensity near

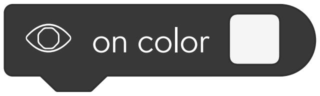
Send Messaging: Sends a message on a given IR channel, from 0 to 7, at a set intensity, from one to 64. Intensity is proportional to proximity, where one is the closest, and 64 is the farthest. You can't overlap sending messages with broadcasting, following, or evading behaviors on the same channels.

Events



on message 9 ▾ received

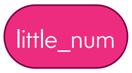
On Message Received: Conditional logic called when an infrared message is received on the specified channel.



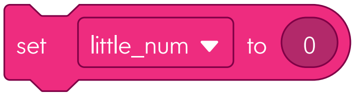
on color □

On Color: Conditional logic called when the color sensor reads a given RGB value, defined as 0 to 255 for the red, green, and blue channels. The color for this block is set using the color sensor on the Sphero RVR+.

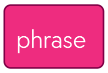
Variables



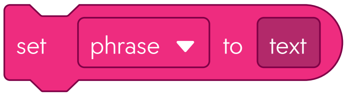
Number Variable: A stored number value that can be assigned and operated on. Values can be integers (whole numbers) or floating point values (numbers with decimals).



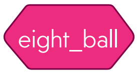
Set Number: Sets the number value for the selected number variable.



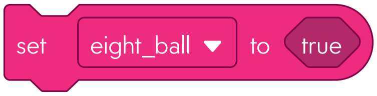
String Variable: A stored string value that can be assigned and operated on. Values can be words or sentences, such as "Hello World!"



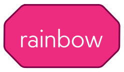
Set String: Sets the string value for the selected string variable.



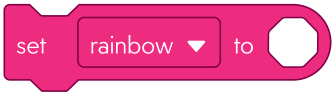
Boolean Variable: A stored boolean value that can be assigned and operated on. Values can be true or false.



Set Boolean: Sets the boolean value for the selected boolean variable.

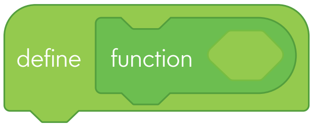


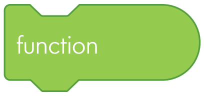
Color Variable: A stored color value that can be assigned and operated on. Values are made up of Red, Green and Blue channels that range from 0 to 255.



Set Color: Sets the color value for the selected color variable.

Functions





Function: Functions allow you to define a reusable group of blocks and then call those blocks from anywhere within the main program. Parameters are like variables, but they are local to the function and help quickly change inputs without modifying the code.